

QtoPCuvDistance.txt

功能：求平面内一点(qx, qy)到B样条曲线P(u)的距离与垂足。

输入参数：(qx, qy)-平面内一点；B样条曲线P(u)由受保护成员控制顶点m_aVertex与节点矢量m_aNode定义。

输出参数：u-垂足在曲线p(u)上的参数，即距离方程(15.1)的解参数。dist-(qx, qy)点到B样条曲线p(u)的距离；(px, py)-垂足。

调用函数：zbrak-用逐步扫描法求曲线p(u)定义域[0, 1]中的缩小了的有根区间和有根区间的数目；rtsafe-牛顿-拉斐森法求连续函数方程在有根区间内的一个近似根；这两个函数由参考文献[212]改造而来，其中用到两个自编函数：Func-求距离方程左端函数值与funcd-求距离方程左端函数值与一阶导数值；GetDBPr-求B样条曲线P(u)上点与导矢，见7.6.4节。

```
double QtoPCuvDistance(double qx, double qy, double &u, double &dist, double &px,
                        double &py)
{
    int n, nbmax, nb, i;
    double x1, x2, px0, py0, dist0;
    CArray<double, double> xb1, xb2;
    xb1.SetSize(21);
    xb2.SetSize(21);
    x1=0.;
    x2=1.;
    n=100;
    nbmax=20;
    nb=nbmax;
    zbrak(qx, qy, x1, x2, n, xb1, xb2, nb);
    double xacc=0.000001*(xb1[0]+xb2[0])/2.;
    double root=rtsafe(qx, qy, xb1[0], xb2[0], xacc);
    GetDBPr(0, Degree, root, px, py);
    dist=sqrt((px-qx)*(px-qx)+(py-qy)*(py-qy));
    for(i=1; i<=nb; i++)
    {
        xacc=0.000001*(xb1[i]+xb2[i])/2.;
        root=rtsafe(qx, qy, xb1[i], xb2[i], xacc);
        GetDBPr(0, Degree, root, px0, py0);
        dist0=sqrt((px0-qx)*(px0-qx)+(py0-qy)*(py0-qy));
        if(dist0<dist)
        {
            u=root;
            dist=dist0;
            px=px0;
            py=py0;
        }
    }
    return(dist);
}
```